

# An Introduction to YOUnite

If you are new to MDM you will need to be familiar with the concepts on the page [What is Master Data Management \(MDM\)?](#).

YOUnite is a robust, configurable, federated data-integration system with master data capabilities. YOUnite allows an organization to gently transition to the goal of MDM without forcing them into an "all-or-nothing" approach. It provides tools that allow data governance to:

- Describe data models
- Group applications/services into organizational boundaries
- Manage access control to the data (governance)
- Decide which applications/services contain master data (*Adaptor Capabilities Lists, or ACLs*)
- Integrate and transform data between applications/services

YOUnite includes a user-facing tool, YOUnite UI, that gives you the ability to secure and control data flow both into and out of your company's departments or divisions. With YOUnite you can create *zones* to group the resources for a given entity. The zones can mirror your company's business structure for a more natural use of the data. For example, an educational institution may have a college district office responsible for many colleges within the district. You can create a "parent zone" for the college district office for accessing data across all of its district colleges. Each college within that district could then be grouped into a "child zone," with YOUnite securing and controlling data flow into and out of each. Read more on [zones](#) below.

YOUnite addresses the issue of "What is the record of truth?" for a resource in any given context. That source of truth is called a *Master Data Record (MDR)*. The resource might be a customer, a product order, a college, a course, etc. MDM provides multiple points for updating these resources. It also ensures that the master data record is always updated while allowing each zone to determine where those updates are applied. For example, if a student updates their address in an application system, YOUnite receives this change and updates the address data record. The college zone's *Adaptor Capabilities Lists (ACLs)* then determine if the updated address is applied to the Student Information System, Learning Management Systems, or any other application that stores the student's address.

## Data Domains

Data domains are the heart of MDM. In traditional database parlance, data domains are a collection of fields (values) that are encompassed by an attribute (database column). For example, using a Customer table example below, the *timeZone* attribute has a data domain of A, P, M, C, E, or null, which represent Alaska, Pacific, Mountain, Central and Eastern time zones. In other words, the data for *timeZone* is limited to this data set, or data domain.



The *timeZone* data for customer records might appear as in the following example:

The *timeZone* data domain:

Customer	custID	firstName	lastName	street	city	state	zip	phone	timeZone
	1	Harry	Jones	11 Main St.	Fairbanks	AK	99701	9078881111	A
	2	Mike	Smith	22 Independence Ave.	Fairfax	VA	22030	5712224444	E
	3	Eliza	Jones	500 Lilac Lane	Chico	CA	95926	5301113333	P

In MDM, domains are schemas that refer to a specific table (i.e. the domain would be Customer, using the example above). The actual data stored (such as each row in the example above) represents what is called the data record.

An MDM data architect's goal is to:

- create data domains that will normalize data across many systems
- manage access to their organization's disjointed data sets (referred to as governance)

Some MDM solutions have fixed or rigid data domains that are generally tied to a given vertical (e.g. manufacturing or finance) but YOUnite allows data architects to:

- create data domains that reflect their organization's requirements or its unique organizational structure (referred to as **multi-domain MDM**).
- version their data domains to accommodate new applications and application versions

Once a domain is created and its data loaded it can be referenced by other data domains, data stewards, and by API consumers as a source of truth. YOUnite domains are defined with JSON.

## YOUnite Data Store vs Federated Domains

Each record within the domain is referred to as a **data record**.

Data records can be stored in either a "YOUnite Data Store" domain (inside YOUnite) or in a federated domain.

1. If a domain is stored inside the MDM (YOUnite) Data Store, then the data records are stored inside YOUnite and can be retrieved, in whole or in part, by applications. Domains that are either not sensitive to an organization (e.g. a list of courses in a college system) or are reference material that doesn't generally change (e.g. list of states, countries, etc) are good candidates for MDM data store domains.
2. If a domain is stored inside a federated domain, the data records are NOT stored inside YOUnite but are created in real-time by referencing the elements (or properties) as they are stored in the company's multiple systems. An example of a good candidate for a federated domain is if one division vigorously guards its data but the entire company would benefit from the sharing of the updated data. For example, suppose the company's sales division spends a great deal of its time maintaining and managing sensitive customer data such that they would not be comfortable migrating their customer data to an MDM data store. With a federated domain, the data would remain in the sales division's system for them to manage and maintain and the company as a whole would benefit from the shared updated customer data. This is a candidate for a federated domain.

In the federated domain model, YOUnite stores information about where the customer data is stored, when it was last updated, and which zone it is shared with, but the customer data itself is physically located only in that organization's various systems. The data stored by YOUnite in this scenario is extraneous information and is referred to as the **metadata**. The permission-related information (which zone the data is shared with) is called governance and is implemented with ACLs. Federated MDM has added latency that is often offset by the way it handles updates to data records. Changes to the data records can be thought of as diffs. Instead of sending the entire data record metadata to YOUnite when a change is made to its data, only the actual changes are sent.

## Domain Property Types

Standard primitives types (integer, floating point numbers, strings, and enumerations) are supported as well as arrays and cross-references between domains.

For more a more on domains see the [Data Domains](#) guide.

---

## The Operational Side of YOUnite: Zones

One of YOUnite's key design solutions is the ability to group an organization's master data resources by the organization's structure (e.g. divisions, departments, districts, schools, etc.) and create relationships between these groups. YOUnite calls these groupings **zones**. An example might be found in a college school system. For instance, YOUnite could mirror the California Community College (CCC) system structure where the zones could be the CCC Chancellor's Office, Los Rios Community College District, and Butte College.

Zones contain domains, adaptors, logs, ACLs, and other resources. Zones are associated with each other in a hierarchical structure with parent, sibling, and child zones. In the example of the college school district, the schools within that district might be considered child zones of the district (parent zone).



## Zone Users

When a zone is created, two distinct users are defined:

1. **Zone Admin:** The administrator is responsible for operational tasks for that zone, such as:
  - Setting policies
  - Creating subordinate zones
  - Creating users
  - Granting API Access
  - Managing operational notifications that are sent and received from other zones
  - Managing and viewing operational logs and notifications
  - Managing operational permissions between zones (i.e. allowing users of other zones to perform the responsibilities listed above)
2. **Zone Data Steward:** The data steward:
  - Has access rights to all of the master data relevant to the zone
  - Has control over the domains and master data associated with a zone.

- Controls inbound and outbound ACLs for permissions to the data
- Creates users that have access to the data
- Can limit or expand user access to the data
- Grant API access to the data
- Manage data related notifications that are sent and received from other zones
- Manage and view domain and master data logs
- Manage inbound/outbound ACLs to master data
- Manage other metadata related to the master data

By default, the zone admin and the zone data steward privileges are mutually exclusive. The zone admin controls the operational aspects of the zone while the zone data stewards controls the data.

YOUnite works with Single Sign-On (SSO) services. When a zone is created, the SSO IDs for the zone's admin and zone data steward must be provided. The same SSO ID can be used for both the zone admin and the zone data steward, creating a user that has control of both the operational AND data elements of a zone (if it is necessary to overrule the default setting).

**NOTE:** As a security measure, the administrator of a parent zone does not have access to its child zones by default.

You can add additional users (individually or as part of a group) to a zone and grant them permissions. YOUnite's permissions model involves **roles** (grouped permissions) and **groups** (grouped users) so that roles can be assigned to groups. This creates an easy-to-manage permission paradigm:

Identity	Description
User	A user in the YOUnite system that is tied to an SSO ID.
Groups	A group contains multiple users.
Permissions	Specifies access or denial to operations and resources.
Roles	A grouping of permission settings.

For more a more on zones see the [Zones, Users, Groups, Roles and Permissions](#) guide.

## Adaptors: The Key to a Federated Ecosystem

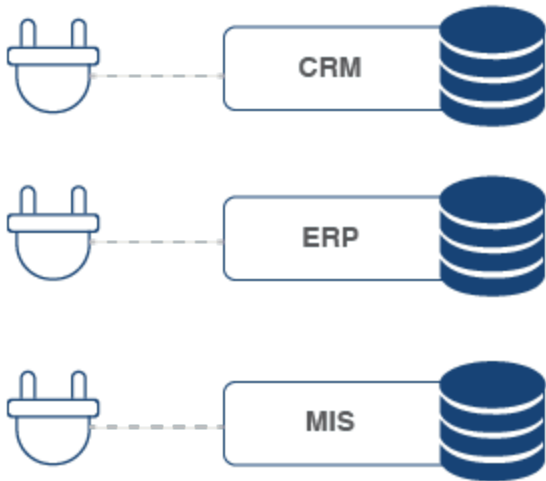
In a federated ecosystem, adaptors are the interface between an organization's disjointed systems. Adaptors perform CRUD (create, read, update, and delete) operations on the data while the YOUnite Data Hub performs the task of distributing (routing) the data between the systems.

Adaptors represent a significant development effort in the overall process of migrating services under master data control.

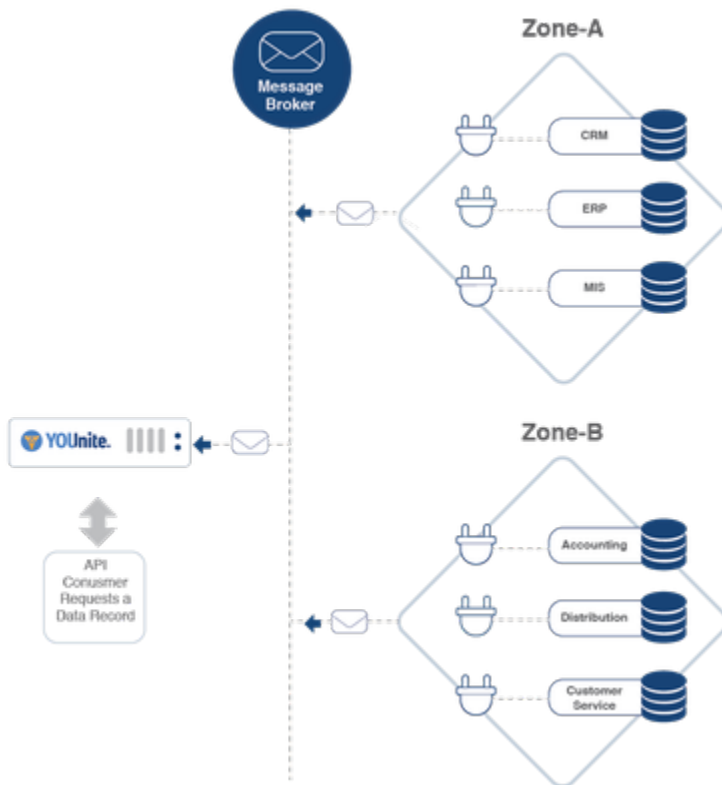
- A more detailed description can be found on the [adaptors](#) page.
- Configuring and managing adaptors can be found on the [Managing Adaptors](#) page.
- The [YOUnite Developer](#) page covers how to use the YOUnite Adaptor SDK for Java to create adaptors.

In the example below, the organization's CRM, ERP, and MIS systems need to be integrated with YOUnite through the adaptors. The adaptors are written if pre-existing adaptors do not exist.





Extending the example above, each adaptor belongs to a zone as in the image below. The ellipse represents a data record being passed between an adaptor or an API consumer to the DataHub and out to other adaptors. The data record is in the form of a JSON object.



## Data Governance: ACLs

ACLs are what allow data visibility between zones and adaptors and are typically managed by the

zone's data steward.

ACLs are a key component of MDM. ACLs are:

- Permissions on out-bound data
- Controls on what in-bound data a zone or adaptor should receive
- Operational permissions: creating data records in the YOUnite ecosystem and mapping entities in source systems to the YOUnite data records.

ACLs can be thought of as a series of filters that get applied to a data operation.

For example, if an update (PUT or PATCH) operation is performed on data under MDM control, ACL filters would control:

- What zones and adaptors would have visibility to the change
- What zones and adaptors want to receive the change

---

## Notifications

YOUnite can notify applications within an organization's ecosystem when:

- operational changes occur within YOUnite, or
- when records under master data control are added, updated, or removed

This allows business processes to take action when certain events occur and provides a mechanism to make applications and services MDM-aware.



---

## Putting it All Together

Once zones, users, data domains, and adaptors are defined, the federated YOUnite runtime ecosystem can be defined into three layers. It's often easiest to understand them from the bottom up.

### Layer 3 - API Access, Data Access, and Notifications

Users and applications can make requests (e.g. governance) and gain data access through the YOUnite API. YOUnite becomes an operation data store when accessing data records through the API.



**Note:** The **YOUnite API** is a RESTful application programming interface that allows developers to access administrative resources and view/modify data records under control of the YOUnite Data Hub. The API should be used for any business logic while the message bus is used for keeping data synchronized.

Notifications allow events to be generated and delivered to legacy applications to trigger business logic. If, for example, an employee is promoted, and the HR system updates the employee's record, the adaptor attached to the HR system can detect the change and pass it on to YOUnite where it can then notify other systems that have registered interest in employee status changes.

---

### Layer 2 - Data Governance

Data governance defines what data a zone chooses to share and receive

(inbound, outbound, and operational ACLs) with other zones and adaptors. For example, the HR Zone may choose to restrict changes it receives (inbound) from a system that is part of the Manufacturing zone. Or, the HR Zone may restrict changes it receives from an entire zone representing a company spin-off subsidiary and all its systems. Or, the HR zone may apply governance ACLs on its own system, preventing personal information from being shared outside of the system (outbound).



---

### Layer 1 - Data Synchronization

The adaptor makes the connections between YOUNite and the legacy systems in the organization's ecosystems. Adaptors extract, transform, and load data into and out of the legacy systems. Business logic is generally not part of an adaptor.

